

دانشگاه صنعتي امیرکبیر  
(پلی تکنیک تهران)

دانشكده علوم کامپیوتر و ریاضی

پروژه سوم یادگیری ماشین

نگارش

سید حسین محمدی

مدرس

دکتر اکبری

آذر 1401

|  |  |
| --- | --- |
| فهرست مطالب | صفحه |

[بخش اول: سؤالات تشریحی 3](#_Toc122019064)

[1-1-1 نحوه دقیق ساخت مدل naïve bayes برای متن 3](#_Toc122019065)

[1-1-2 نحوه آزمایش مدل 3](#_Toc122019066)

[1-1-3 منطقی بودن فرض استقلال شروط 3](#_Toc122019067)

[1-2 هسته های معتبر 4](#_Toc122019068)

[1-3 اجرای NB بروی نمونه 4](#_Toc122019069)

[1-4 امتیازی هوفدینگ 4](#_Toc122019070)

[1-5 امتیازی 6](#_Toc122019071)

[بخش دوم: تمرین مطالعه (Reproducing Kernel Hilbert Space) 7](#_Toc122019072)

[2-1 مقدمه 7](#_Toc122019073)

[2-1 تعریف فضا 7](#_Toc122019074)

[2-2 اهمیت این فضا 8](#_Toc122019075)

[بخش سوم: پیادهسازی 9](#_Toc122019076)

[3-1 پیش پردازش 9](#_Toc122019077)

[3-2 انتخاب بهترین ستون ها 10](#_Toc122019078)

[3-3 مقایسه مدل ها 11](#_Toc122019079)

[3-4 مرز های تصمیم (با TSNE) 13](#_Toc122019080)

[3-5 مدل رای اکثریت 14](#_Toc122019081)

[3-6 تاثیر مقادیر k 15](#_Toc122019082)

[3-1 اضافه – انتخاب ستون متفاوت 15](#_Toc122019083)

[3-1 توضیح مدل 17](#_Toc122019084)

[3-2 نتیجه 17](#_Toc122019085)

# بخش اول: سؤالات تشریحی

## نحوه دقیق ساخت مدل naïve bayes برای متن

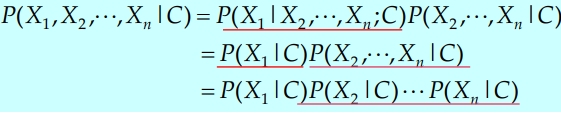
فرض کنیم ویژگی هدف داده ما c1,c2,…ck کلاس دارد. همچنین هر داده ما متشکل از متن هست.

حال داده را براساس کلاس هدف جدا میکنیم. برای کلاس Ci، متن ها را درنظر بگیرید؛ تعداد کلمات مجزا را برای همه متن ها میشماریم و تقسیم بر تعداد کل داده های Ci میکنیم. با اینکار احتمال شرطی دیدن یک کلمه مثل xj درصورت آنکه کلاس ci باشد را بدست میآوریم: P(xj|ci)=#xj form data of ci / #data of ci

حال با کمک از این احتمال ها، با فرض مستقل بودن این مقادیر، میتوان احتمال ظهور کلمات یک متن را برای هر کلاس تشخیص داد.

## نحوه آزمایش مدل

احتمال ظهور کل متن (تمامی کلمات) در با فرض اینکه برای یک کلاس خاص باشد را برای همه کلاس ها حساب میکنیم. فرض میکنیم احتمال ظهور یک کلمه مستقل از کلمات دیگر باشد، در اینصورت:



حال این مقدار را در احتمال ظهور همان کلاس نیز ضرب میکنیم که احتمال اینکه این متن و کلاس درست باشند را به ما میدهد. حال بین مقادیر کلاس ها، آن را که بزرگتر هست بر میگزینیم.

## منطقی بودن فرض استقلال شروط

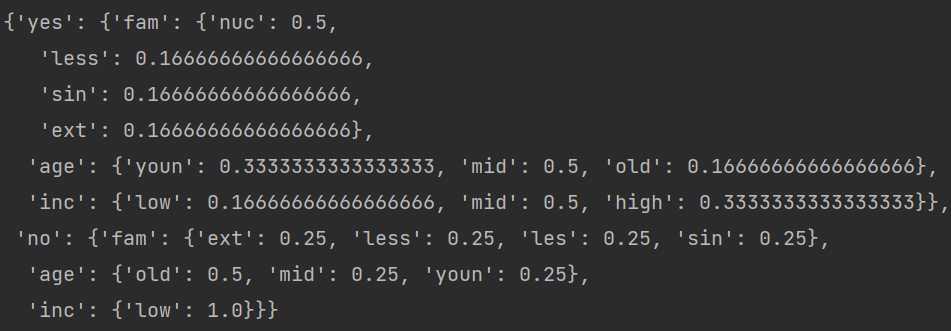
خیر. زیرا که احتمال وجود کلمه ای را میتوان تا حدی بر اساس کلمات قبلی سنجید.

## هسته های معتبر

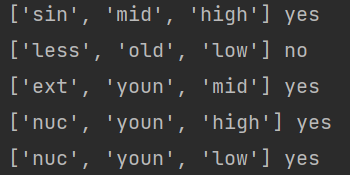
1. *معتبر هست زیرا مجموعه دو ماتریس* Mercer *یک ماتریس مثبت و متغارن میسازد.*
2. *معتبر هست زیرا ضرب دو ماتریس* Mercer *یک ماتریس مثبت و متغارن میسازد.*
3. *معتبر هست زیرا عدد اویل به توان هر عدد مثبت میماند و همجنین تقارن ماتریس نهایی را تغییر نمیدهد.*
4. *معتبر نیست زیرا لزوما برای هر مقداری مثبت نمیماند.*

## اجرای NB بروی نمونه

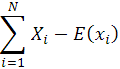
*مدل بدست آمده:*

**

*جواب سوال:*

**

## امتیازی هوفدینگ

*آ. فرض کنید بازه شامل بینهایت نباشد. در آن صورت تنها حالتی که مجموعه فاصله ویژگی ها از میانگین از t (که به بینهایت میرود) بزرگتر باشد باشد ()، آن هست که بینهایت ویژگی وجود داشته باشد. پس احتمال وقوع آن 0 هست. در نتیجه اگر از هردو لوگ گرفته شود، میشود:*

*درست میماند.*

*فرض کنید مدل اول احتمالاتی در ازای ورودی، مقدار a را خروجی میدهد که ½+b درصد اقات درست است (میتوان همان ورودی چندبار اجرا کرد و جواب متفاوت گرفت).*

*حال مدل دومی را میسازیم که ورودی را N بار به مدل لوب میدهد و خروجی های ai را بدست میآورد؛ سپس رای اکثریت بین این N تا خروجی مدل اول میگیرد.*

*تغییر متغیر: برای هر خروجی مدل اول، ai، یک xi در نظر بگیرید که نشان دهندی صحیح بودن اون خروجی هست. به زبان دیگر، اگر xi برابر 1 باشد به معنای صحیح بودن خروجی ai و اگر -1 باشد، برعکس.*

*xi بین یک و منفی یک هست و ½+b درصد اوقات 1 میباشد. میانگین مقادیر ممکن برای xi برابر با 2b میباشد و بازه مقادیر برابر 2 هست. پس E(xi)=2b و M-m=2*

*احتمال آنکه مدل دوم خروجی اشتباه بدهد، برابر است با احتمال آنکه مجموعه xi ها کمتر از 0 باشد هست زیرا که به معنای وجود بیش از N/2تا عدد -1 میباشد.*

*فرض کنید ما تلاش داریم احتمال خطا را به e کاهش دهیم، تنها متغییر موجود N هست. به زبان دیگر:*

*محاسبه دقیق این احتمال دشوار است، میتوان از نابرابری هوفدینگ استفاده کرد ولی ابتدا باید کمی تغییرات بدهیم*

*پس حال فقط باید N را به گونهای برداریم که عباریت راستی کوچکتر از ln e باشد. حال یک معادلی برای N میسازیم.*

## امتیازی

*---*

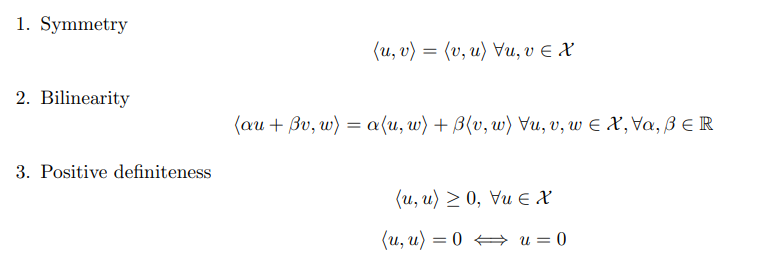
# بخش دوم: تمرین مطالعه (Reproducing Kernel Hilbert Space)

## مقدمه

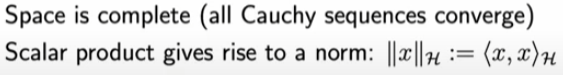
برای نمایش، اثبات و کار کردن با توابع خطی، از فضا های هیلبرت که متشکل از توابع خطی هستند استفاده میشود. فضای Reproducing Kernel Hilbert Space برای آنالیز توابع کرنال استفاده میشوند و خواص مفید بسیاری دارند.

## تعریف فضا

عملیاتی که دو وکتور از یک فضای خطی را گرفته و یک عدد میدهند را در نظر بگیرید. این عملیات در صورت پیروی از خواص زیر، یک نوع **ضرب داخلی** هست (<x,y>=o(x)’o(y)):

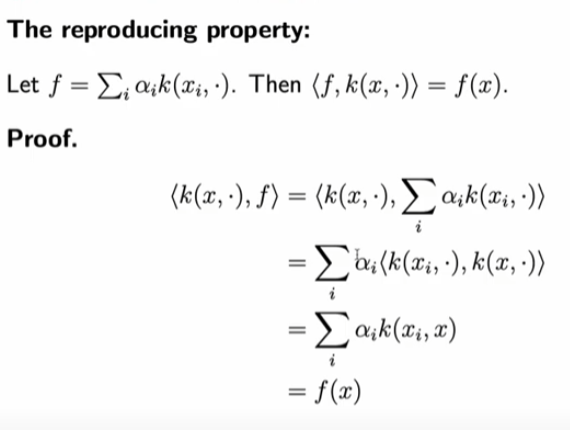


حال برای ضرب داخلی مثل <x,y>، مجموعه تمام وکتور هایی که خواص زیر را دارند را درنظر بگیرید. این مجموعه، **فضای هیلبرت** نام دارد:

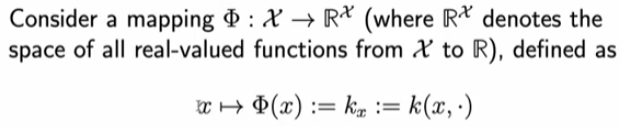


سرانجام اگر ضرب داخلی مذکور، به شکل یک کرنال باشد، آنگاه، فضای هیلبرت آن را، **Reproducing Kernel Hilbert Space** (RKHS) مینامیم.

علت وجود Reproducing در نام این فضا، خاصیت زیر هست:



که در آن  تابع به شکل زیر هست:



## اهمیت این فضا

اثبات اینکه  که به این معنا هست که k یک کرنال هست، اگر و فقط اگر تابعی وجود داشته باشد که ضرب داخلی آن تابع با مقدار کرنال برابر شود را میتوان با کمک از فضای RKHS نشان داد.

همچنین میتوان کلی بودن (universal) یک کرنال را با این فضا نشان داد.

# بخش سوم: پیادهسازی

**3-آ مقایسه مدل ها**

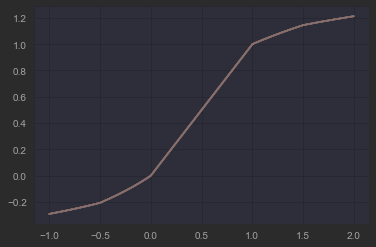
## پیش پردازش

ابتدا کمی داده را تمیزتر و یکپارچه تر کردیم. برای مثال ستون income دو کلاس باید باشد ولی به دلیل خطا در تایپ، 4 تا بوده.

چون هدف ما یافتن کلاس خلاف قاعده است، حذف و مقادیر نادر و یا outlier ممکن هست باعث آسیب به مدل شود. به همین دلیل داده را تغییر میدهیم و حذف نمیکنیم.

ابتدا مقادیر کمیاب در ستون های کتگوریکال بود. برای مثال برای هر ستون، تمامی مقادیری که کمتر از 0.3% تکرار شدند، یا 0.6% تکرار شدند اما نسبت تعدادشان برای دو کلاس شبیه بودند را با مقدار other جایگزین کردیم.

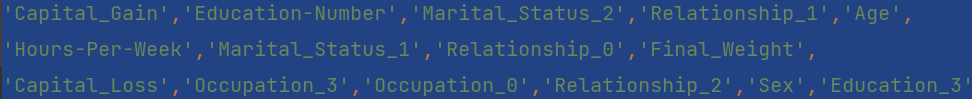
سپس مقادیر outlier بازه ای که از حداکثر 99% داده بیشتر اند (یا حداقل کمتر از 1%) را به میانگین نزدیک میکنیم. برای اینکار از تابع با شکل زیر، مقادیر خارج از بازه 0-1 را به 0.5 نزدیک میکنیم. (شبیه به x^1/p). اینکار تاثیر بدی روی KNN میگذارد زیرا که فاصله ها تغییر میکنند اما چون برای کمتر از 2% داده انجام میشود مشکلی نیست. (کوچکترین کلاس بیش از 24% هست)

****

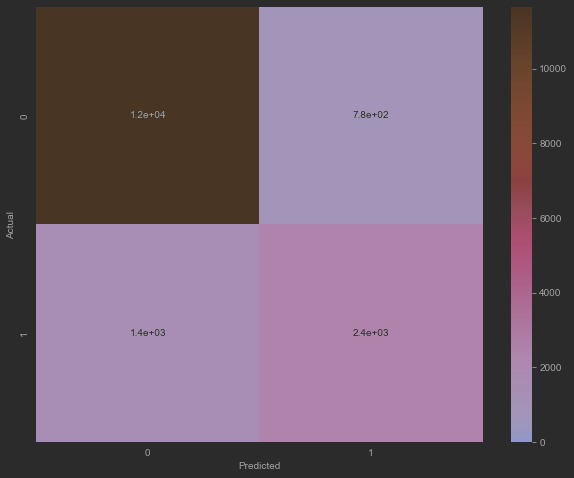
حال داده های کتگوریکال را به دلیل نیاز KNN و SVM به مفهوم فاصله، با binaryencoding تبدیل به چند ویژگی مستقل میکنیم. علت انتخاب binary این هست که ستون ها با یکدیگر همبستگی کمتری دارند. اما برای Naïve Bayes شاید بهتر باشد آن را قرار نداد.

## انتخاب بهترین ستون ها

با کمک از جنگل تصادفی بهترین ستون ها را انتخاب میکنیم. این هر درخت به تنهایی میتواند بهترین ویژگی را بردارد. با جنگل، میتوان ویژگی هایی را انتخاب کرد که با هم همبستگی کمتری دارند که برای NB مهم هست.



همچنین، این جنگل، یک تخمین برای بهترین دقت روی تست ممکن را نیز به ما میدهد.



## مقایسه مدل ها

مدل SVM به شدت کند هست و دقت خوبی ندارد. برای تسریع پروسه یافتن هایپر پارامتر، از فقط ¼ داده استفاده میشود.



عملکرد بروی تست:



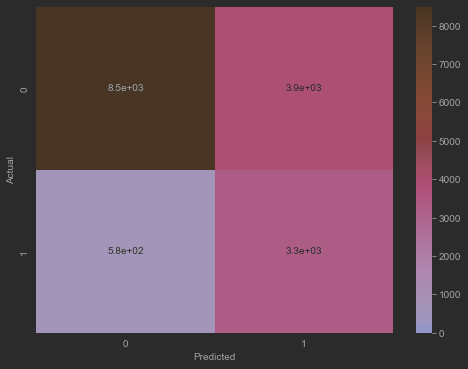
مدل KNN سرعت خیلی بهتری دارد. با اینکه KNN به حجم داده اهمیت نمیدهد پس نیاز به تغییر در حجم آن نیست و هایپر پارامتر بهینه برای آن  است.

عملکرد بروی تست:



مدل NB از همه سریعتر عملکرد داشت اما نتیجه مطلوبی نداشت. این مدل به تعداد هر کلاس اهمیت میدهد، پس ، برای اطمینان، به دلیل نابرابر داده، مقداری از داده های کلاس کمتر به داده به صورت رندوم اضافه شدند. هایپرپارامتر بهینه آن فرقی با مقدار عادی ندارد. همچنین بین گزینه ها، ComplementNB بهترین نتیجه را داشته.

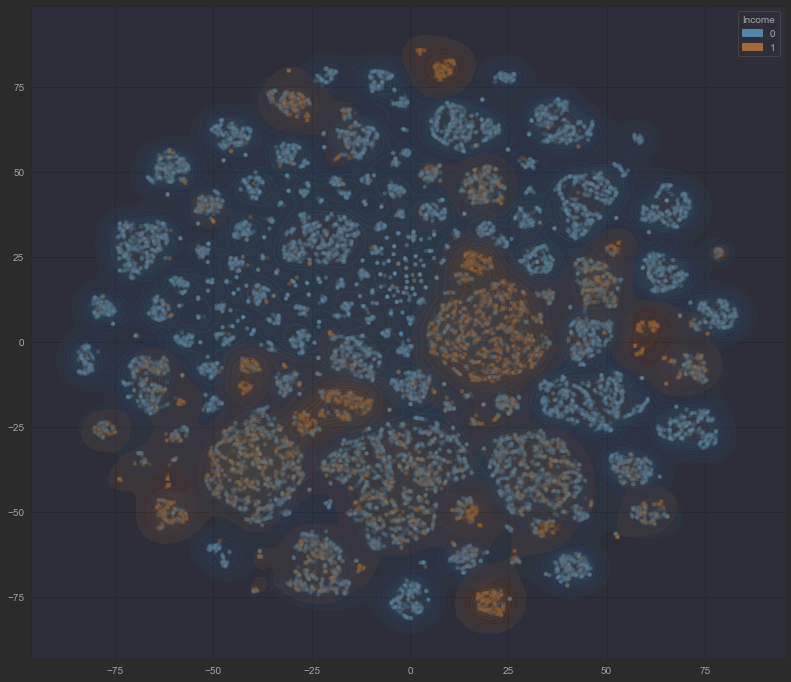
عملکرد بروی تست:



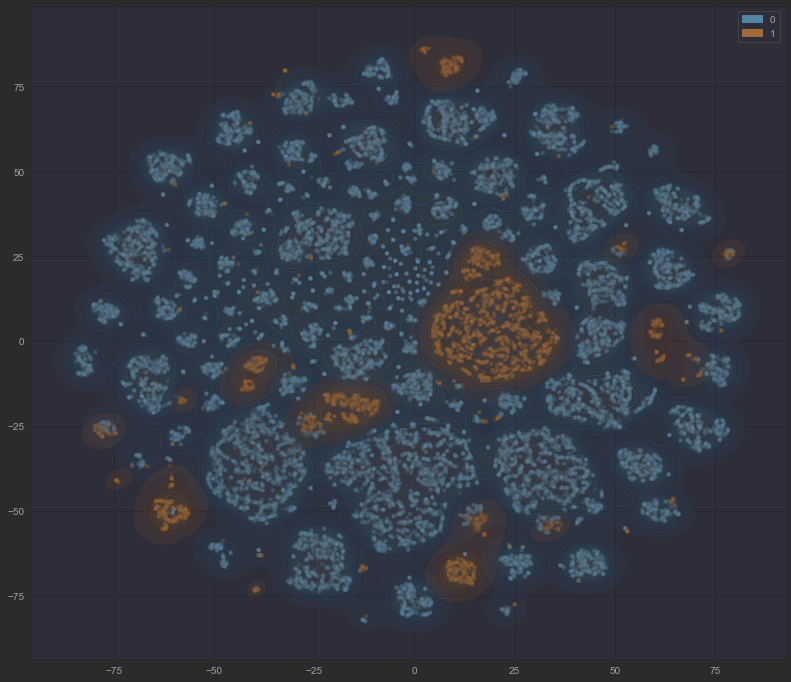
## مرز های تصمیم (با TSNE)

داده چند بعدی را نمیتوان نمایش داد، به همین دلیل با کمک از TSNE آن را به 2 بعد میبریم. سپس مرزهای تصمیم آن را برای مدل های مختلف رسم میکنیم.

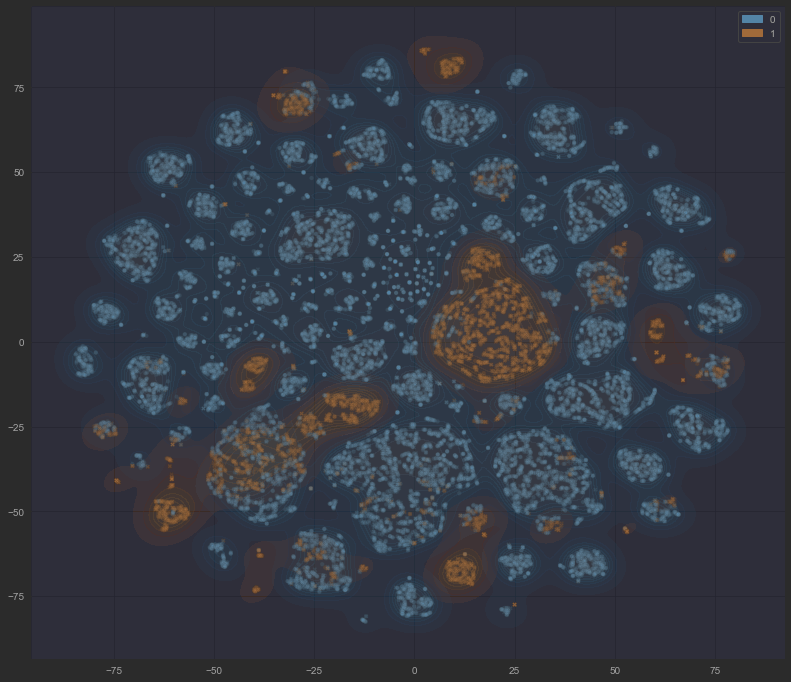
مقدار واقعی:



SVM



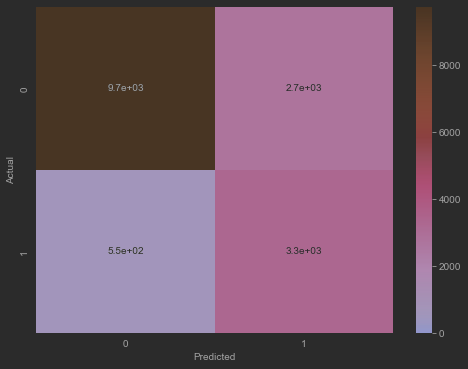
KNN

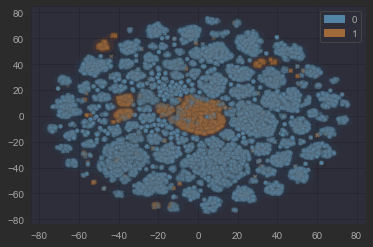


## مدل رای اکثریت

انتخاب مدل: Svm بدلیل عملکرد بد (سرعت و دقت) در نظر گرفته نشده، مدل NB به دلیل سریع بودن، به مدل اضافه میکنیم به همراه 2 KNN با 2 مقدار متفاوت برای k. با افزایش k، دقت تست بهتر و دقت تمرین کمتر میشود. به همین دلیل یکی را روی 25 و دیگری را روی 50 قرار میدهیم.

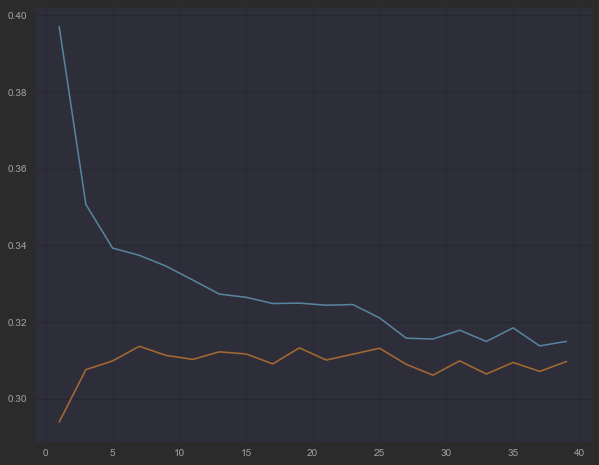
از لحاظ اجرا از SVM بدتر هست و عملکرد قابل قبولی دارد. از لحاظ تمرین از svm سریعتر میباشد.





## تاثیر مقادیر k

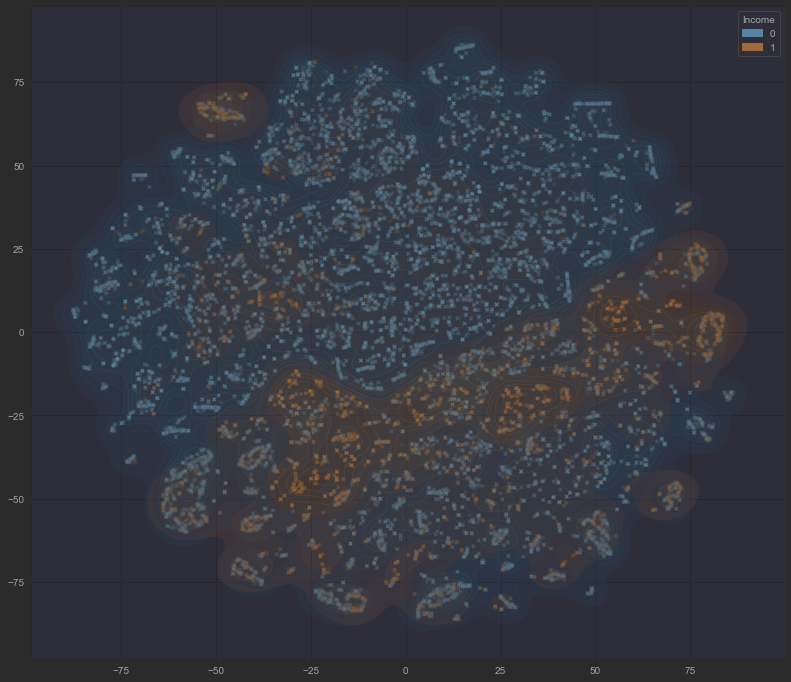
به دلیل کندی اجرا، برای مقادیر فرد انجام شده. همچنین فقط ½ داده با در نظرگیری لیبل داده شده است.



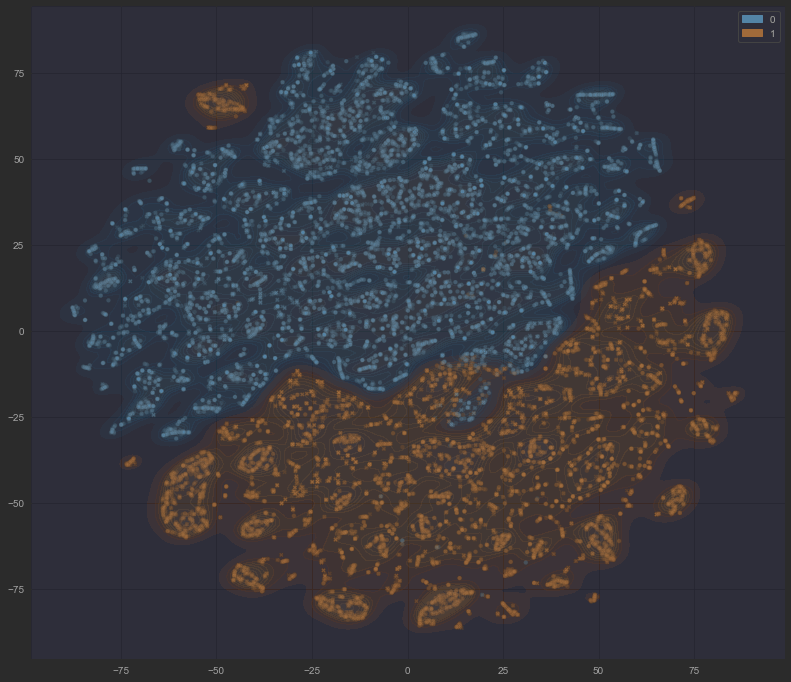
## اضافه – انتخاب ستون متفاوت

در بالا، همبستگی بین ستون ها را تغییر ندادیم. روشی مثل naïve bayse نسبت به این همبستگی به مشکل میخورند. به همین دلیل تست های بالا را یک بار با انجام PCA (بدون حذف ستون) بر روی داده و انتخاب بهترین ویژگی ها با کمک از درخت انجام دادیم. عملکرد مدل های دیگر کمی بدتر شد اما به دلیل اینکه داده خروجی PCA، همبستگی بین ستونی پایینتری دارند، عملکرد NB کمی بهتر شد. همچنین، مرز تصمیم آنها بعدر از TSNE راحتر قابل تشخیص هست.

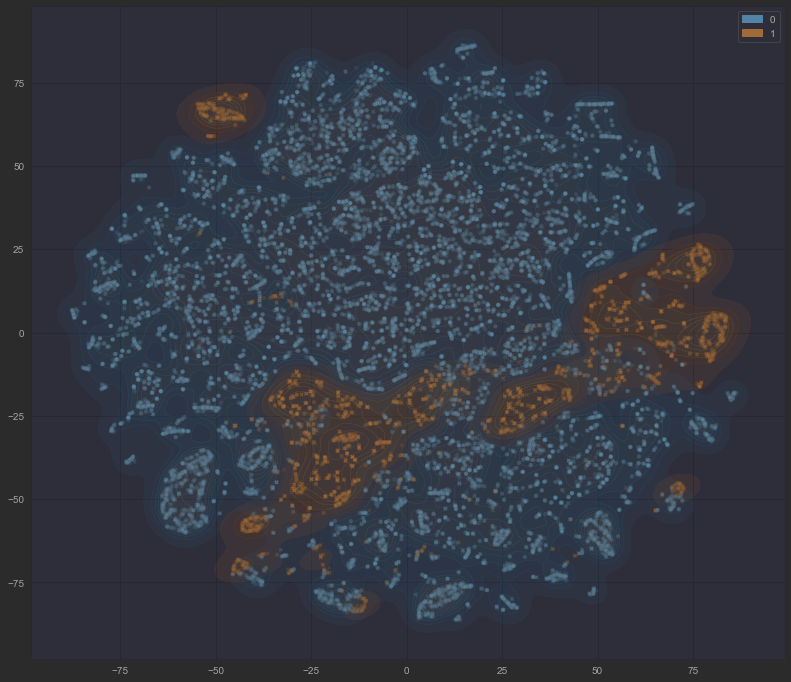
مقادیر واقعی:



NB:



KNN:



**3-ب DESlib**

## توضیح مدل

یک مدل ترکیبی از متشکل از مدل KNN هست.

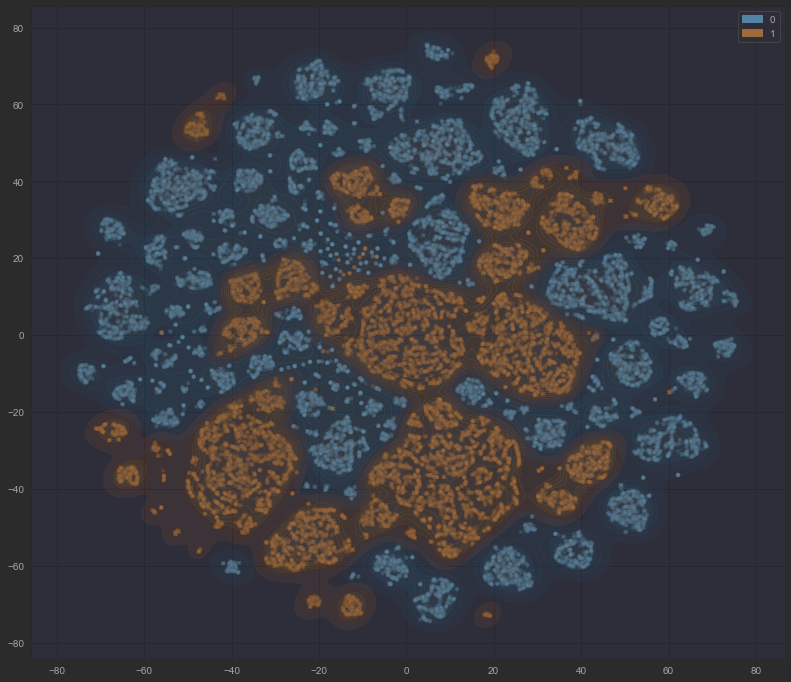
ابتدا M مدل ساخته میشوند. N تا از بهترین ها از نظر دقت انتخاب میشوند. سپس J تا از آن Nتا بر اساس متنوع تر بودن مدل انتخاب میشود تا مدل نهایی ایجاد شود.

## نتیجه

از لحاظ تمرین و آموزش، نتیجه خوبی دارد و سرعت قابل قبولی دارد.

هایپر پارامتر های بهینه: 







Amirkabir University of Technology  
(Tehran Polytechnic)

Department of Computer Science and Math

Machine Learning Assignment #3

By

Seyed Hossein Mohammadi

Taught by

Dr. Akbari

Oct of 2022